

ABEHZELE Esteban

Classe : GSI-1

FOWDAR Heeshaam

Groupe W

LAM Emilie

MANRESA Mattéo

NEHAD Younes

# Rapport

# Projet JEE

Professeur référent : Mohamed HADDACHE

Matière : Développement JEE

Année scolaire : 2025-2026



## Sommaire

Introduction.....	3
Organisation et méthodologie.....	5
Conception technique.....	7
Problèmes rencontrés et solutions.....	8
Résultats et fonctionnalités.....	9
Conclusion et perspectives.....	10

# Introduction

## Présentation de l'équipe

Ce projet a été réalisé par une équipe d'étudiants motivés, mobilisant leurs compétences en développement web, conception UML et gestion de données. Les membres de l'équipe sont :

- Heeshaam FOWDAR
- Esteban ABEHZELE
- Emilline LAM
- Younes NEHAD
- Mattéo MANRESA

L'équipe s'est appuyée sur une communication continue via des réunions régulières, un partage d'écran lors des sessions de développement, et des échanges asynchrones via des outils collaboratifs. Cette coordination a permis de maintenir une vision commune du projet et d'assurer une répartition équilibrée du travail.

Pour la collaboration et le développement, nous avons utilisé les outils suivants :

**GitHub, Jira, Discord**

**IntelliJ IDEA, MySQL Workbench, WampServer, Tomcat**

## Description du projet

Le projet consistait à développer une application JEE complète permettant de gérer plusieurs aspects internes d'une entreprise :

- **Gestion des employés** : Le projet permet d'effectuer diverses manipulations sur la base de données concernant les employés tels que : Ajout, Modification des informations, Suppression, Listage des employés, Recherche, Affectation à un projet
- **Gestion des départements** : Le projet permet d'ajouter, supprimer, lister les départements. Il faudra aussi pouvoir affecter un département à un employé ainsi que pouvoir lister tous les employés d'un département.

- **Gestion des projets internes** : Le projet permet de créer, modifier et supprimer un projet. De plus, nous pourrons aussi affecter des employés à un projet ainsi que suivre l'état d'avancement du projet.
- **Génération de fiches de paie mensuelles** : Créer une fiche de paie pour un mois donné en faisant le calcul automatique du net à payer. Nous devons aussi pouvoir consulter les fiches de paie d'un employé et pouvoir les imprimer. Finalement nous devons pouvoir rechercher une fiche de paie.
- **Authentification + autorisation par rôle** : Avoir différents accès pour différents rôles.

Le public visé par cette application est restreint, il ne touche que les employés, managers et services RH d'une entreprise quelconque.

L'application repose sur :

- **JSP / HTML / CSS / JS** pour les vues
- **Hibernate + JDBC** pour la persistance
- **MySQL** pour la base de données
- **Tomcat** pour le déploiement des **Servlets** qui représente la couche contrôleur
- **IntelliJ IDEA** comme IDE principal

Une seconde version du projet a été réalisée en **Spring Boot**, suivant une architecture moderne (Spring MVC, Spring Data JPA, Spring Security, Thymeleaf).

## Cahier des charges

Le cahier des charges définissait plusieurs objectifs fonctionnels et techniques :

- Implémenter une **application Web dynamique** respectant l'architecture MVC
- Permettre la gestion complète des ressources RH de l'entreprise
- Assurer la **cohérence et la sécurité** des données
- Intégrer des fonctionnalités avancées comme :
  - Affectation des employés à des projets
  - Calcul automatique du net à payer
  - Statistiques RH et projets
- Proposer une **réécriture du projet avec Spring Boot**
- Fournir :
  - Rapport complet
  - Schémas MCD
  - Code accessible via GitHub
  - Présentation orale

# **1. Organisation et méthodologie**

## **1.1 Répartition des rôles dans l'équipe**

Afin d'organiser efficacement le projet, nous avons choisi d'utiliser **Jira** comme outil principal de gestion des tâches. Dès le début du développement, toutes les fonctionnalités et sous-fonctionnalités décrites dans le cahier des charges ont été décomposées en tickets Jira. Chaque ticket représentait une tâche précise (par exemple : création de la base de données des employés, implémentation du module de paie, développement de l'interface JSP pour la gestion des projets, etc.).

Cette méthode nous a permis :

- **de visualiser clairement l'ensemble des tâches à réaliser,**
- **de prioriser les fonctionnalités** en fonction de leur importance et de leur dépendance,
- **de suivre l'avancement du projet** grâce aux tableaux Kanban et aux sprints définis dans Jira.

Contrairement à une répartition figée des responsabilités, nous avons adopté une approche flexible et collaborative. Chaque membre de l'équipe pouvait consulter la liste des tâches disponibles et choisir librement celles qu'il souhaitait réaliser, en fonction de ses compétences, de ses envies ou de sa disponibilité. Cette liberté a favorisé l'implication de chacun et a permis à l'équipe de rester motivée tout au long du projet.

En pratique, cela signifiait que :

- Les développeurs back-end pouvaient s'orienter vers les tickets liés à la logique métier et aux servlets.
- Les développeurs front-end pouvaient prendre en charge les tickets concernant les interfaces JSP et l'ergonomie.
- Le responsable base de données se concentrait sur les tickets liés au schéma relationnel et à la persistance.
- Le responsable qualité/test choisissait les tickets liés aux tests unitaires, d'intégration et à la validation des fonctionnalités.

Cette organisation souple, soutenue par Jira, a permis de maintenir une transparence totale sur l'avancement du projet et de renforcer la responsabilisation individuelle tout en gardant une cohésion d'équipe.

## 1.2 Outils utilisés

### 1.2.1 Outils de développement

- **IntelliJ IDEA** (IDE principal – développement Java, Servlets, Hibernate)
- **Apache Tomcat** (serveur de déploiement)
- **Postman / navigateur** (tests des endpoints)
- **GitHub** (versioning et travail collaboratif)
- **Visual Studio Code** (édition HTML/CSS ponctuelle)

### 1.2.2 Outils de gestion de base de données

- **MySQL** (SGBD principal)
- **MySQL Workbench** (visualisation, gestion des tables)
- **Wamp Server**
- **Hibernate ORM** (mapping entre entités Java et tables SQL)

## 1.3 Flux de travail

Le workflow adopté :

Notre projet a suivi une approche structurée et itérative, permettant de progresser de la conception

théorique à l'implémentation concrète, tout en assurant une intégration continue des retours et des optimisations. Le flux de travail s'est déroulée selon les grandes étapes suivantes :

### 1. Décomposition des tâches

- Nous avons débuté par la définition des différentes tâches et User Story à faire dans notre backlog afin de détailler précisément le résultat attendu.

### 2. Phase de Conception Préliminaire :

- Nous avons ensuite défini l'architecture logicielle, ce qui a nécessité l'élaboration d'un diagramme de classes afin de représenter les principales entités du labyrinthe ainsi que leurs relations.

### **3. Développement des Composants Métier :**

- Après la phase de conception, nous nous sommes concentrés sur l'implémentation de la base de données
- Simultanément nous nous sommes concentrés sur la création des entités fondamentales, à savoir les Servlets, les modèles et les DAO.
- Cette étape a été suivie par le développement de la logique applicative, assurant la gestion des interactions entre les différentes couches et pages.

### **4. Construction des Interfaces Utilisateur :**

- Une fois les fonctionnalités métier consolidées, nous avons entamé la construction de l'interface utilisateur à l'aide de pages JSP, permettant une interaction dynamique via le web.

### **5. Migration vers SpringBoot :**

- Après la version JEE classique, nous avons migré l'application vers Spring Boot afin de moderniser l'architecture et simplifier le déploiement. Cette migration a rendu l'application plus modulaire, plus maintenable et mieux adaptée aux standards actuels du développement web.

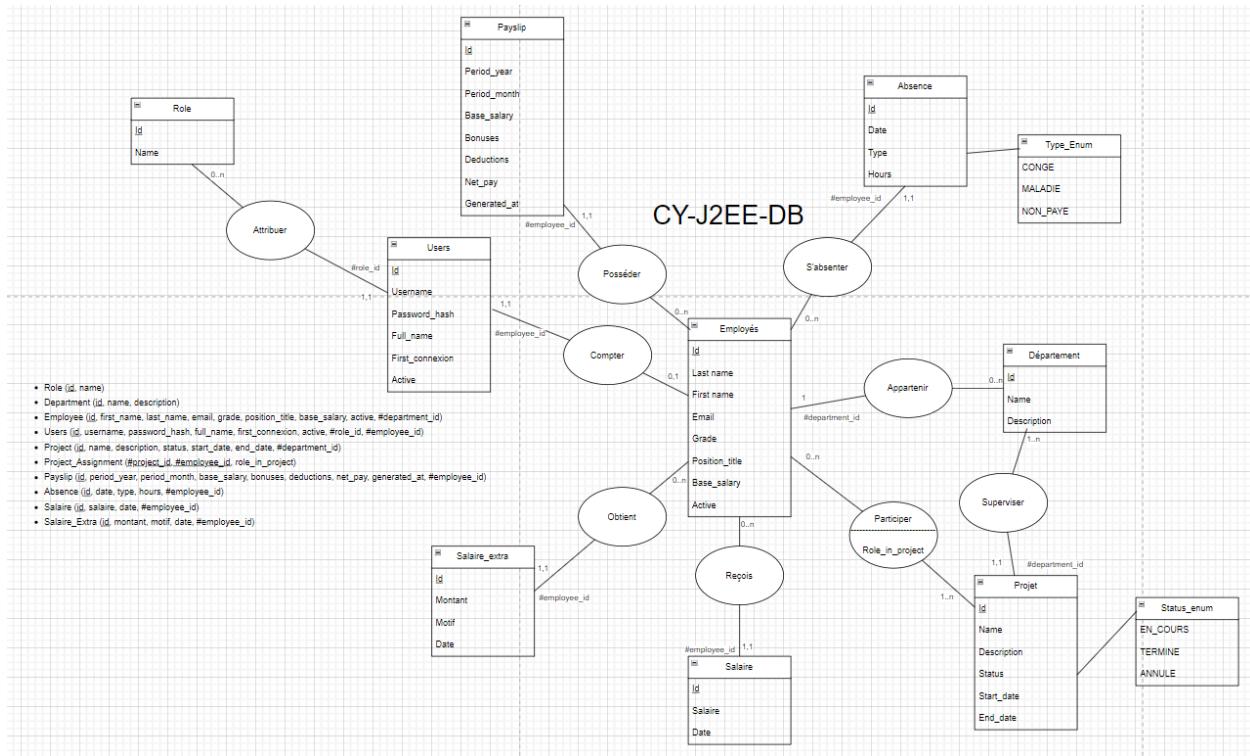
### **6. Finalisation et Documentation :**

- Un fichier README détaillé a été rédigé pour faciliter la prise en main du projet par d'autres développeurs ou utilisateurs.
- Enfin, la préparation du présent rapport a été initiée.

Ce flux de travail nous a permis de gérer la complexité du projet en décomposant les tâches et en assurant une progression méthodique vers l'atteinte de nos objectifs.

## 2. Conception technique

MCD:



MLD :

- Role (`id`, name)
- Department (`id`, name, description)
- Employee (`id`, first\_name, last\_name, email, grade, position\_title, base\_salary, active, #department\_id)
- Users (`id`, username, password\_hash, full\_name, first\_connexion, active, #role\_id, #employee\_id)
- Project (`id`, name, description, status, start\_date, end\_date, #department\_id)
- Project\_Assignment (#project\_id, #employee\_id, role\_in\_project)
- Payslip (`id`, period\_year, period\_month, base\_salary, bonuses, deductions, net\_pay, generated\_at, #employee\_id)
- Absence (`id`, date, type, hours, #employee\_id)
- Salaire (`id`, salaire, date, #employee\_id)
- Salaire\_Extra (`id`, montant, motif, date, #employee\_id)

ACTIVE

## **3. Problèmes rencontrés et solutions**

### **3.1 Difficultés techniques**

1. Gestion des sessions utilisateurs et rôles
2. Validation des formulaires (Servlet côté serveur)
3. Synchronisation JSP → Servlets (params manquants ou mal nommés)
4. Calcul automatique des fiches de paie
5. Déploiement sur Tomcat + erreurs compilation web.xml

### **3.2 Solutions mises en œuvre**

- Standardisation des noms de champs + DTO
- Ajout de filtres Servlet pour la sécurité
- Utilisation d'un **Service Layer** pour séparer la logique métier
- Revue complète des relations SQL → Hibernate

## **4. Résultats et fonctionnalités**

### **4.1 Fonctionnalités implémentées**

#### **Gestion des employés**

- **CRUD complet** : Il est possible d'ajouter, modifier, consulter et supprimer les fiches des employés. Chaque opération est sécurisée et validée côté serveur.
- **Recherche avancée** : Un moteur de recherche permet de filtrer les employés par nom, prénom, matricule, grade, poste ou département, facilitant l'accès rapide à l'information.
- **Affectations multiples** : Un employé peut être affecté à plusieurs projets simultanément, avec un rôle spécifique pour chaque affectation. Cette flexibilité reflète les réalités du travail en entreprise.

#### **Gestion des départements**

- **Création / mise à jour** : Les départements peuvent être créés, modifiés ou supprimés par les administrateurs.
- **Liste des membres** : Chaque département affiche dynamiquement la liste de ses membres, avec leurs postes et grades, facilitant la supervision.

#### **Gestion des projets**

- **Suivi d'avancement** : Chaque projet possède un statut (en cours, terminé, annulé) et des dates de début/fin. L'interface permet de visualiser l'état d'avancement en temps réel.
- **Affectations des employés** : Les employés peuvent être affectés à un ou plusieurs projets, avec un rôle défini (développeur, chef de projet, etc.).
- **Suppression sécurisée (intégrité référentielle)** : La suppression d'un projet est conditionnée par l'intégrité référentielle : les affectations sont automatiquement mises à jour ou supprimées pour éviter les incohérences.

#### **Gestion des fiches de paie**

- **Génération automatique** : Les fiches de paie sont générées automatiquement à partir du salaire de base, des primes et des déductions.
- **Export imprimable** : Chaque fiche peut être exportée au format imprimable, facilitant la distribution ou l'archivage.

- **Calcul net = base + primes – déductions** : Le calcul est effectué selon la formule : Net = Salaire de base + Primes - Déduction . Les déductions peuvent inclure les absences non rémunérées ou les avances sur salaire.

## Authentification / autorisation

- **Admin** : Accès complet à toutes les fonctionnalités, y compris la gestion des utilisateurs et des rôles.
- **Employé** : Accès restreint à ses propres données (fiche de paie, affectations, absences). La gestion des sessions et des mots de passe est sécurisée, avec vérification des identifiants et contrôle des permissions.

## Version Spring Boot

- **Architecture modernisée** : Utilisation de Spring Boot pour simplifier la configuration, améliorer la modularité et faciliter le déploiement.

## BONUS : Gestion des absences

- **Générer une absence pour un employé** : Il est possible d'enregistrer une absence (congé, maladie, non payé) pour un jour donné.
- **Lister toutes les absences** : Une vue dédiée permet de consulter l'historique des absences de tous les employés.
- **Filtrer la liste des absences (Par employé)** : Les absences peuvent être filtrées par employé, type ou période, facilitant le suivi et le calcul des déductions.

## 4.2 Captures d'écran

Cette section illustre une partie du fonctionnement de l'application à travers un scénario de démonstration.

- Connexion en tant qu'admin
- Création d'un employé
- Recherche dans la liste
- Création d'un projet
- Affectation du projet à l'employé
- Suppression de l'employé

### Connexion

**Nom d'utilisateur**  
admin

**Mot de passe**  
\*\*\*\*\*

**Se connecter**

**Prénom**  
Jean

**Nom**  
Dujardin

**Email (auto-généré)**  
jean.dujardin<id>@entreprise.com

**Grade**  
Employee

**Poste**  
Administrateur RH

**Salaire de base**  
2000

**Date d'embauche (jour/mois/année)**  
10 10 2024

**ID Département**  
Ressources Humaines

**Retour** **Créer l'employé**

ID	Arrivée	Nom	Email	Département	Grade	Poste	Actions	
2	2025-06-15	Dupont Jean	jean.dupont2@entreprise.com	Ressources Humaines	Senior	Responsable RH	<span>Modifier</span> <span>Générer fiche</span>	<span>Fiches de paie</span> <span>Ajout extra</span>
3	2024-10-10	Dujardin Jean	jean.dujardin3@entreprise.com	Ressources Humaines	Employee	Administrateur RH	<span>Modifier</span> <span>Générer fiche</span>	<span>Fiches de paie</span> <span>Ajout extra</span>

**Rechercher et filtrer**

duja

Grade : -- Poste : -- Département : --

Rechercher Réinitialiser

ID	Arrivée	Nom	Email	Département	Grade	Poste	Actions	
3	2024-10-10	Dujardin Jean	jean.dujardin3@entreprise.com	Ressources Humaines	Employee	Administrateur RH	<span>Modifier</span> <span>Générer fiche</span>	<span>Fiches de paie</span> <span>Ajout extra</span>

### Ajouter un projet

**Nom du projet**  
Projet1

**Description**  
Projet

**Date de début**  
07 / 11 / 2025

**Date de fin**  
27 / 11 / 2025

**Département**  
Ressources Humaines

**Status :**  
En cours

[Retour](#) [Créer le projet](#)

## Gestion des projets

### Liste des projets

ID Projet	Nom	Description	Date début	Date fin	Département	Status	Actions
1	Projet1	Projet	2025-11-07	2025-11-27	Ressources Humaines	EN_COURS	<a href="#">Membres</a> <a href="#">Modifier</a> <a href="#">Affecter un employé</a> <a href="#">Supprimer</a>

[+ Ajouter un projet](#) [Retour au tableau de bord](#)

### Affecter un employé à un projet

#### Nouvelle affectation

**Employé**  
Dujardin Jean

**Rôle dans le projet**  
Manager

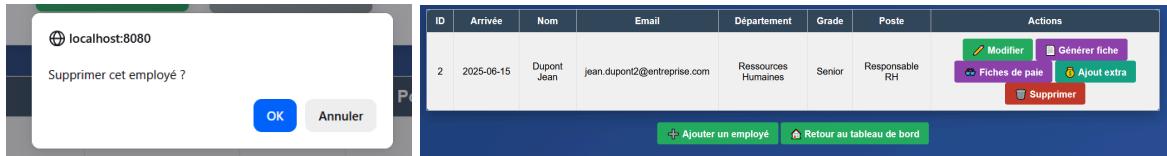
[Retour](#) [Affecter](#)

## Membres du projet

### Liste des membres

ID Projet	Nom	Email	Role	Actions
3	Jean Dujardin	jean.dujardin3@entreprise.com	Manager	<a href="#">Retirer</a>

[Retour à la liste des projets](#)



## 4.3 Tests et validation

La robustesse et la fiabilité de l'application ont été garanties par une série de tests rigoureux menés tout au long du développement. Nous avons systématiquement vérifié le bon fonctionnement de l'application sur différentes plateformes, notamment Windows et Ubuntu, en conditions réelles d'utilisation. Les tests ont été effectués à la fois dans l'environnement de développement IntelliJ IDEA et via le navigateur web, en simulant les interactions des différents profils utilisateurs (administrateur, RH, chef de projet, employé).

Des campagnes de tests quotidiennes ont permis d'identifier rapidement les anomalies potentielles, qu'il s'agisse de dysfonctionnements dans les formulaires, de problèmes d'intégrité des données ou de comportements inattendus lors des opérations critiques (génération de fiches de paie, affectation à des projets, suppression d'entités liées).

L'ensemble de ces validations techniques confirme que l'application est stable, performante et prête à être déployée dans un environnement professionnel.

# **Conclusion et perspectives**

## **Bilan du projet**

L'application JEE développée répond efficacement au cahier des charges : gestion complète des ressources humaines, interface fonctionnelle, base de données cohérente, sécurité implémentée, et migration vers Spring Boot réussie.

Tout au long du projet, l'équipe a su faire preuve d'une organisation rigoureuse et d'une communication harmonieuse, favorisant une progression constante et cohérente. Les obstacles techniques rencontrés ont été relevés avec méthode et persévérance, grâce à une mobilisation collective et une capacité d'adaptation face aux imprévus.

La fiabilité de l'application a été confirmée par une série de tests réguliers menés sur différentes plateformes, garantissant sa stabilité et sa compatibilité. Ce projet illustre pleinement la réussite d'une démarche structurée dans la conception et le développement d'une solution logicielle ambitieuse, répondant à des besoins métiers complexes avec efficacité et précision.

L'équipe a acquis une expérience solide en :

- Architecture MVC
- Servlets / JSP
- Hibernate
- MySQL
- Spring Boot + Spring Security